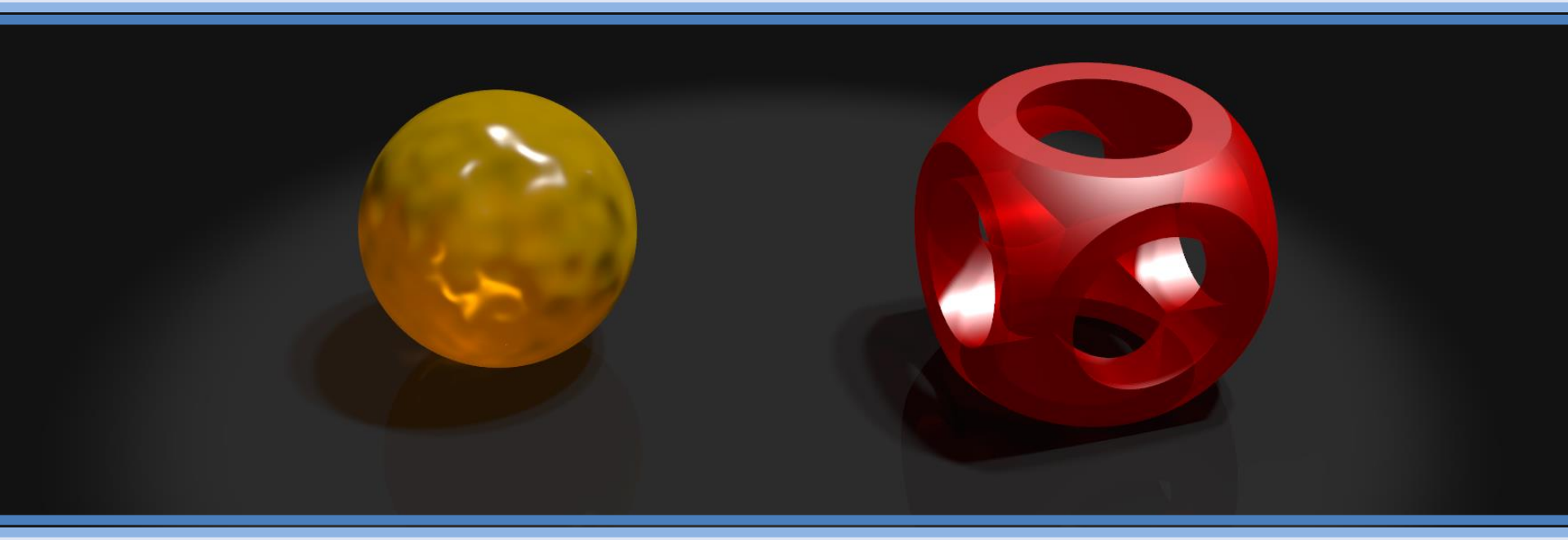
Computergrafik

T. Hopp

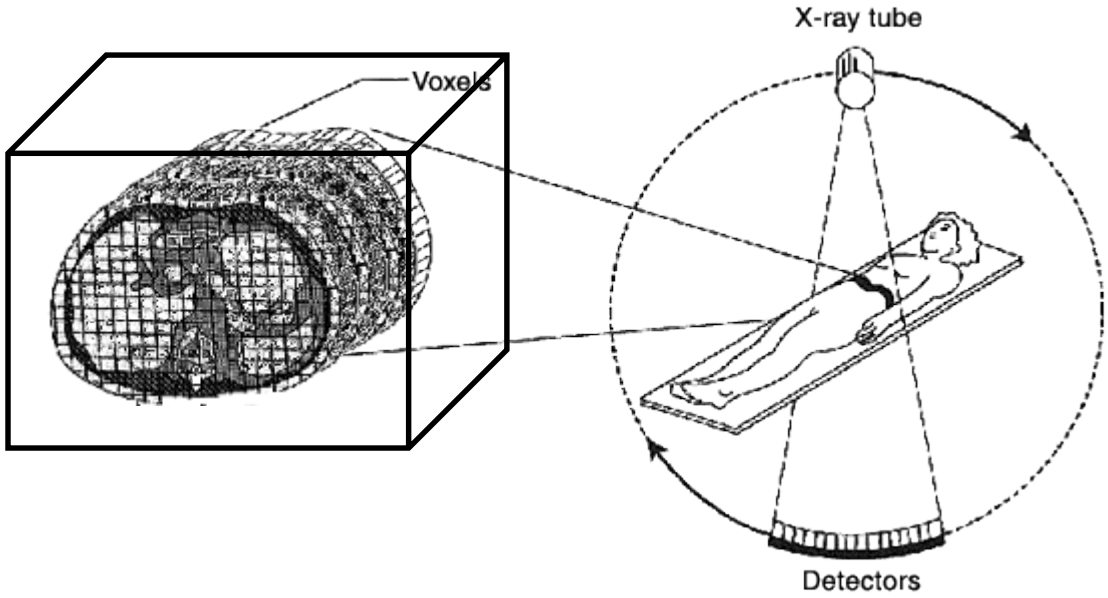


Themenübersicht

1. Einführung
2. Programmierbibliotheken / OpenGL
3. Geometrische Repräsentation von Objekten
4. Koordinatensysteme und Transformationen
5. Zeichenalgorithmen
6. Buffer-Konzepte
7. Farbe, Beleuchtung und Schattierung
8. Texturen
9. Animationen
10. Raytracing
11. **Volumenvisualisierung**

Volumendaten

- Viele bildgebende Verfahren erzeugen 3D Volumendaten
- Oft in diskretem kartesischem Gitter angeordnete Voxel
- Zur Darstellung auf dem Bildschirm muss die Information auf 2D reduziert werden

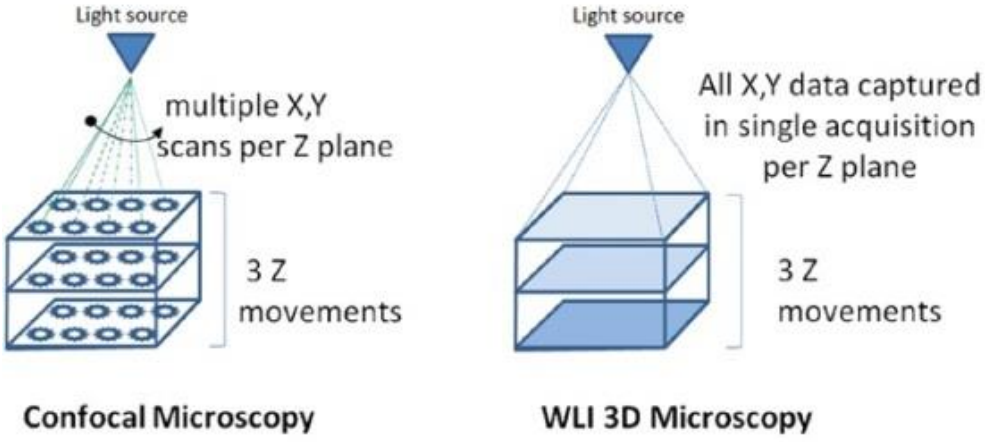


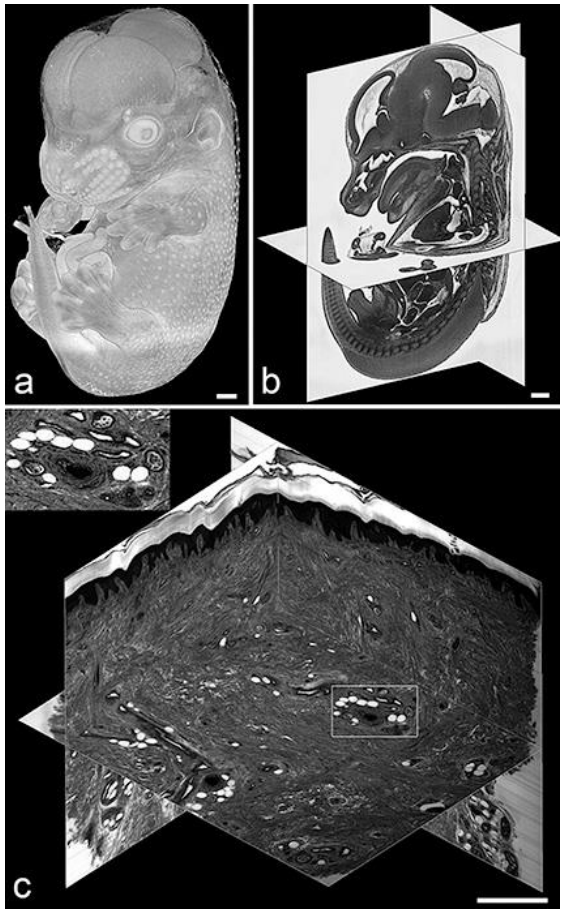
Beispiel: Computertomographie

http://www.physicscentral.com/explore/action/images/scans-img2.gif

Volumendaten

- Z.B. 3D Mikrospkopie



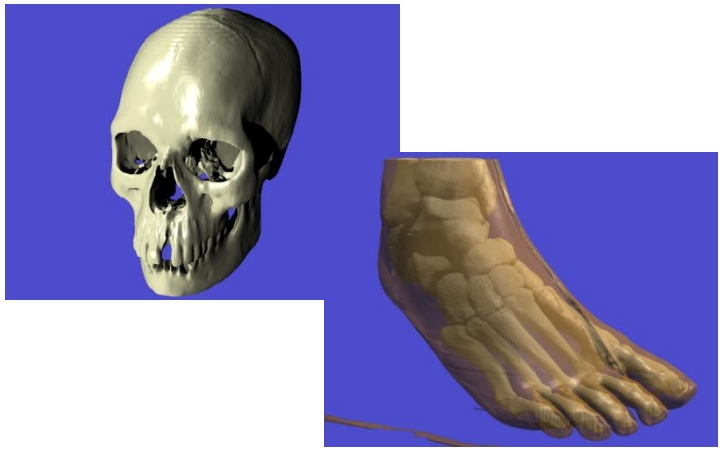


  - Extrem hohe Auflösungen, bis Teravoxel!
  - Oft Zeitreihen (4D)

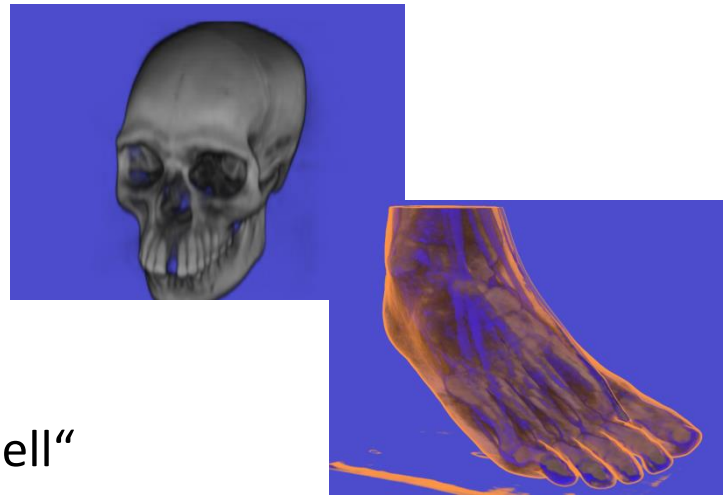
http://www.azom.com/article.aspx?ArticleID=10612
http://www.imaging-git.com/science/light-microscopy/hrem-high-resolution-episcopic-microscopy

Indirekte ⇔ Direkte Volumenvisualisierung

- Indirekte Volumenvisualisierung:
  - Extraktion einer Oberfläche aus Volumendaten, z.B. Isofläche
  - Siehe Polygonisierung!

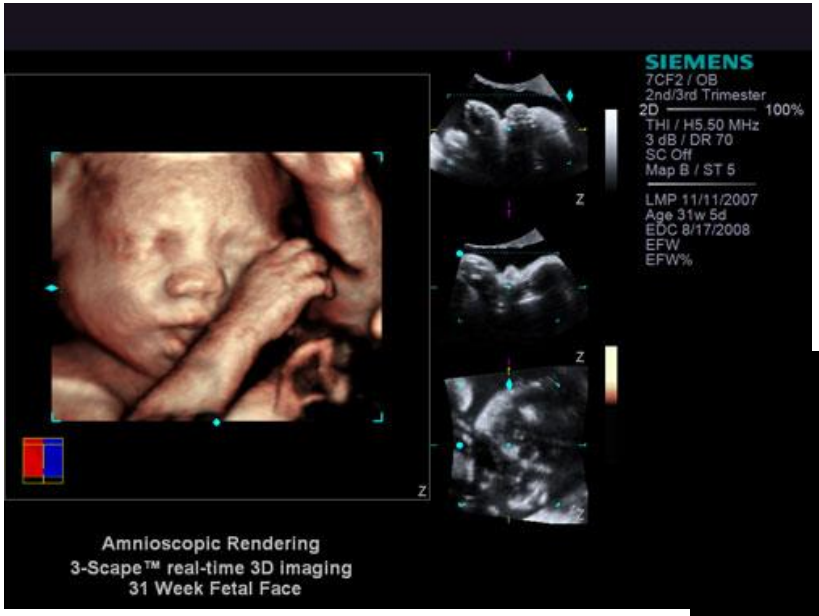


- Direkte Volumenvisualisierung:
  - Betrachtung des Intensitätswerts jedes einzelnen Voxels
  - Verwendung eines optischen Modells: Übersetzung des Intensitätswertes in physikalische Eigenschaften
  - Daraus Beschreibung der Interaktion des Lichts an jedem Voxel → „Optisches Modell“

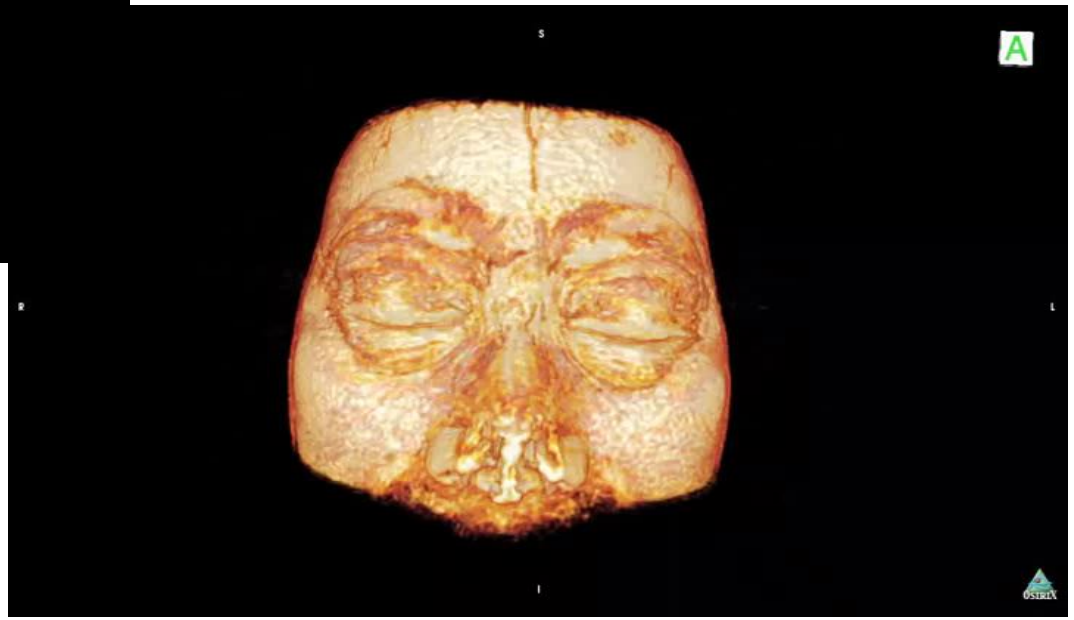


http://idav.ucdavis.edu/~okreylos/PhDStudies/Spring2000/ECS277/DataSets.html

Direkte Volumenvisualisierung in der Medizin



3D-Ultraschall in der Schwangerschaft



Visualisierung eines Aneurysmas aus MRT-Daten

http://www.nvidia.de/object/io_1259596736392.html
https://www.youtube.com/watch?v=v1fdoabjCWk

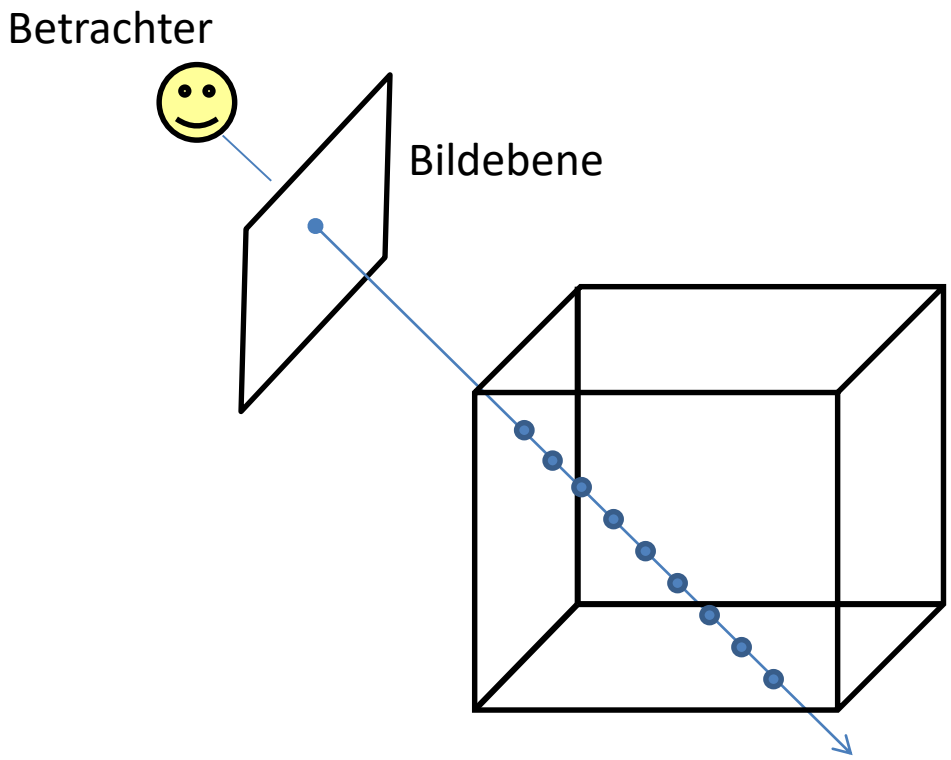
11.1 VOLUME RENDERING INTEGRAL

Optische Modelle

- Optisches Modell basiert auf Konzept des Strahlungstransportes: Modellierung des Verhaltens von Licht, das durch das Volumen läuft.
- Vereinfachung für die Volumenvisualisierung: Volumen = Ansammlung licht-emittierender und -absorbierenden Kugeln (=Voxel)
- Die wichtigsten Modelle sind:
  - **Nur Absorption:** Kugeln absorbieren nur Licht
  - **Nur Emission:** Kugeln emittieren nur Licht
  - **Absorption + Emission:** Kugeln emittieren Licht und absorbieren Licht, das auf sie trifft. Keine Streuung und Reflexion
  - **Streuung + Schattenwurf:** Teilweise Streuung des Lichtes an Kugeln, Schattenwurf auf verdeckte Kugeln

Grundidee der direkten Volumenvisualisierung

- Sichtstrahlen von jedem Pixel der Bildebene durch das Volumen
- Voxel des Volumens liefern definierte optische Dichtewerte (Emission, Absorption)
- Summation entlang der Sichtstrahlen



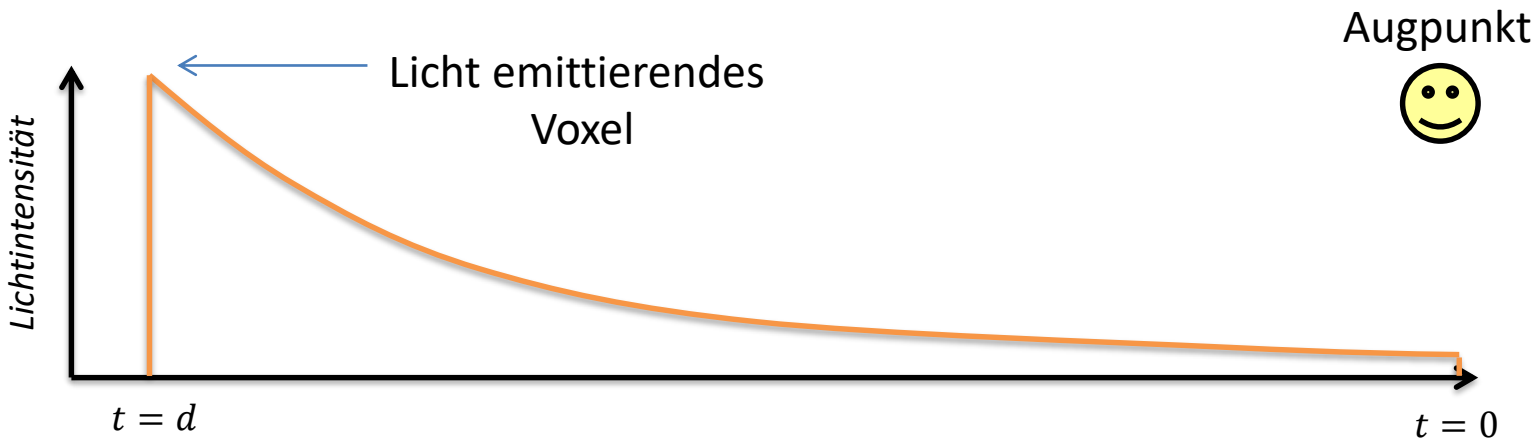
Volume Rendering Integral

- Volumenvisualisierungsalgorithmen lösen i.d.R. das Volume Rendering Integral (… mit mehr oder weniger Approximationen)
- Annahme für Herleitung: Volumen und Abbildung der optischen Dichte sind kontinuierlich.

- Definitionen

$\vec{x}(t)$	Strahl vom Augpunkt in das Volumen, t ist die Distanz zum Auge
$s\ (\vec{x}(t))$	Intensitätswert an der Position $\vec{x}(t)$
$c(t) \coloneqq c(s\ (\vec{x}(t)))$	Emission
$\kappa(t) \coloneqq \kappa(s\ (\vec{x}(t)))$	Absorption

Volume Rendering Integral



- Licht wird auf dem Weg von der Quelle zum Auge kontinuierlich absorbiert
- Einfachster Fall: κ konstant ⇒ von einem Voxel ausgestrahlte Intensität die am Augpunkt ankommt:

$$c^* = c \cdot e^{-\kappa d}$$

- I.d.R wird κ nicht als konstant angenommen, d.h. Absorption ist vom Raumpunkt abhängig:

$$c^* = c \cdot e^{-\underbrace{\int_0^d \kappa(t)\, dt}_{\text{= optische Tiefe}}}$$

Volume Rendering Integral

- Im Auge kommt nicht nur emittiertes Licht eines Voxels an, sondern von allen Voxeln!
- Dies wird abgebildet durch Integration über alle Positionen auf dem Sichtstrahl. Es ergibt sich das gesamte eintreffende emittierte Licht C

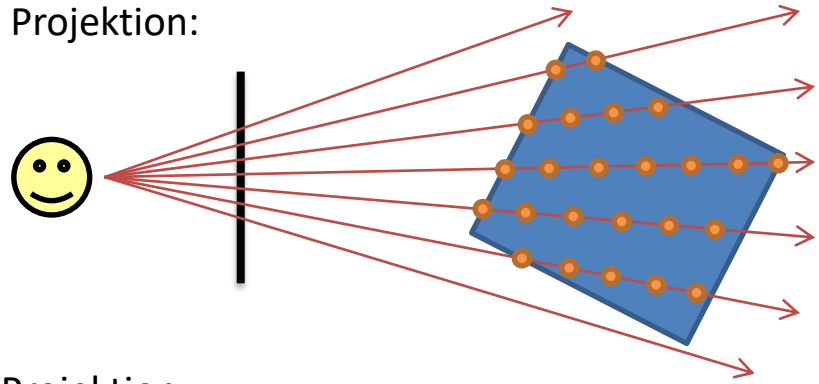
$$C = \int_0^\infty c(t) \cdot e^{-\int_0^d \kappa(\hat{t})\, d\hat{t}}\, dt$$

Volume Rendering Integral

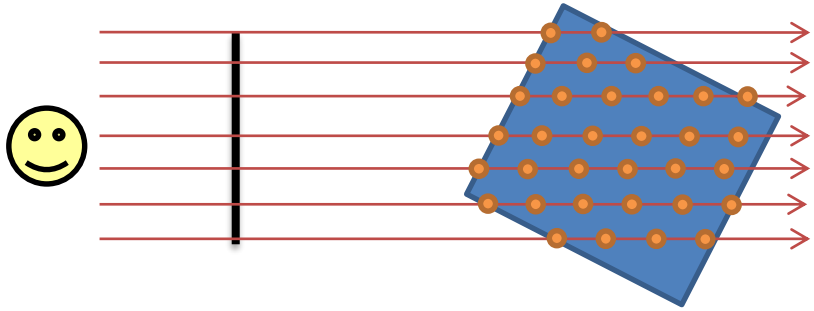
11.2 RAY CASTING

Ray Casting

- Numerische Lösung des Volume Rendering Integrals
- Von jedem Bildpunkt in der *near clipping plane* ausgehend Verfolgung eines Sichtstrahls durch das Volumen.
- Startpunkt der Sichtstrahlen wird durch Kameraeinstellungen bestimmt:
  - Perspektivische Projektion:



  - Orthografische Projektion:



Ray Casting

- Approximation des Volume Rendering Integrals durch Riemann-Summe zur Diskretisierung.
- Exponent des Volume Rendering Integrals:

$$-\int_0^t \kappa(t) \approx -\sum_{i=0}^{t/\Delta t} \kappa(i \cdot \Delta t) \cdot \Delta t$$

- Damit ändert sich das Volume Rendering Integral zu:

$$\tilde{C} = \int_0^\infty c(t) \cdot e^{-\sum_{i=0}^{t/\Delta t} \kappa(i \cdot \Delta t) \cdot \Delta t} \, dt$$

$$\tilde{C} = \int_0^\infty c(t) \cdot \prod_{i=0}^{t/\Delta t} e^{-\kappa(i \cdot \Delta t) \cdot \Delta t} \, dt$$

$$\tilde{C} = \sum_{i=0}^{n} c_i \cdot \prod_{j=0}^{t/\Delta t} e^{-\kappa(j \cdot \Delta t) \cdot \Delta t}$$

Ray Casting: Alpha Blending

- Die Berechnung des diskreten Volume Rendering Integrals kann nun als iterative ***Blending Operation*** zwischen den Abtastpunkten entlang eines Sichtstrahls beschrieben werden.
- Opazität A_j eines Abtastpunktes approximiert dessen ***Absorption***
- Intensität c_i eines Abtastpunktes approximiert dessen ***Emission***
- Mit $A = 1 - e^{-\int_0^d \kappa(t)dt}$ (= Opazität, Alphawert) ergibt sich die Näherungsformel

$$\tilde{C} = \sum_{i=0}^{n} c_i \cdot \prod_{j=0}^{\frac{t}{\Delta t}} 1 - A_j$$

Ray Casting: Alpha Blending

Zwei Strategien für die Berechnung:

- **Back-to-Front**:
  - Iterative Aufsummation von Volumenende zu Bildebene

$$c'_i = c_i + (1 - A_i)c'_{i+1}$$
$$c'_n = 0$$

- **Front-to-back:**
  - Iterative Aufsummation von Bildebene zu Volumenende

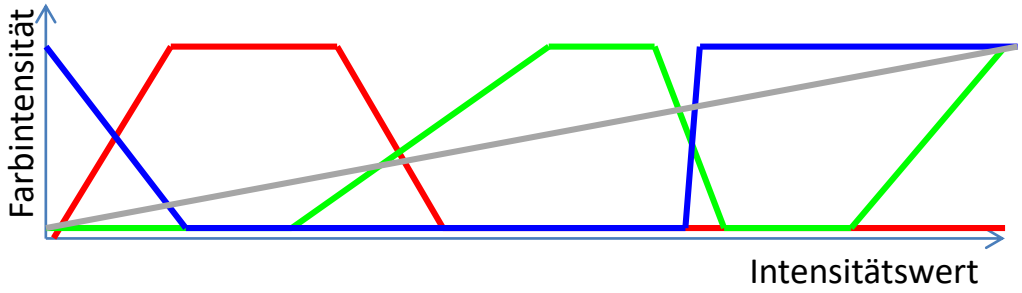
$$c'_i = c'_{i-1} + (1 - A'_{i-1})c_i \qquad c'_0 = 0$$
$$A'_i = A'_{i-1} + (1 - A'_{i-1})A_i \qquad A'_0 = 0$$

  - Erfordert neben Berechnung des Farbwertes c auch die Verfolgung des Opazitätswertes A: zusätzlicher Rechenaufwand
  - Vorteil wenn entlang eines Sichtstrahls vorzeitig abgebrochen werden kann, da z.B. bereits volle Deckkraft erreicht wurde

Ray Casting: Transferfunktion

- Abtastung des Volumens an diskreten Punkten entlang des Sichtstrahls
  - Trilineare Interpolation in $2 \times 2 \times 2$ Zelle
- Aufsummation der ermittelten Werte (vgl. Volume Rendering Integral)
  - Back-to-Front oder Front-to-Back
  - → Intensitätswert pro Pixel in der *near clipping plane*

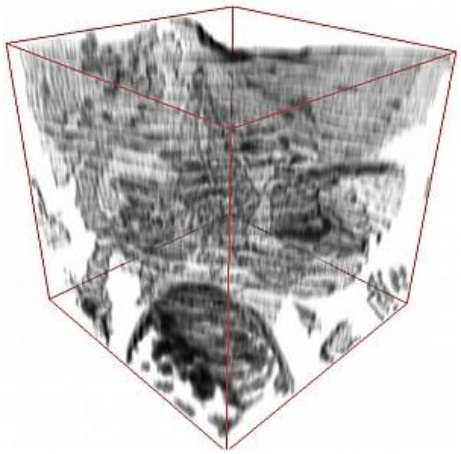
- Mapping der Intensitätswerte mit Transferfunktion: RGB und Alpha-Werte



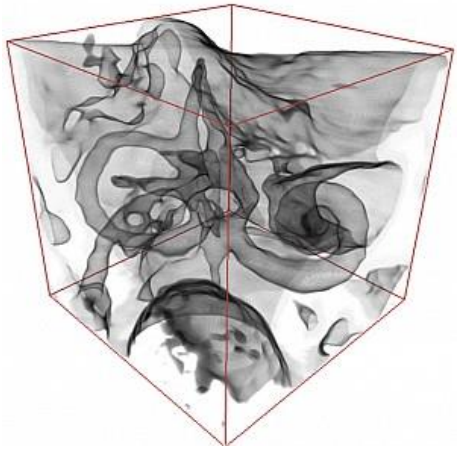
R, G, B-Kanal
+ Alpha-Kanal

Transferfunktion: Prä-/Postklassifizierung

- Prä-Klassifizierung
  - Vor Interpolation Umsetzung der Intensitätswerte des Volumens in Farbwerte und ggf. eine Opazität
- Post-Klassifizierung
  - Umsetzung des skalaren Summenwertes nach Interpolation in einen Farbwert und ggf. eine Opazität



Prä-Klassifizierung



Post-Klassifizierung

http://schorsch.efi.fh-nuernberg.de/roettger/index.php/VolumeRendering/Pre-AndPost-Classification

Ray Casting: Beleuchtung

- Zusätzlich Beleuchtung der Szene möglich (vgl. Beleuchtungsmodelle)
- Berechnung der Normalen für Volumendaten durch Approximation mittels Gradient:
  - Im kontinuierlichen Fall:

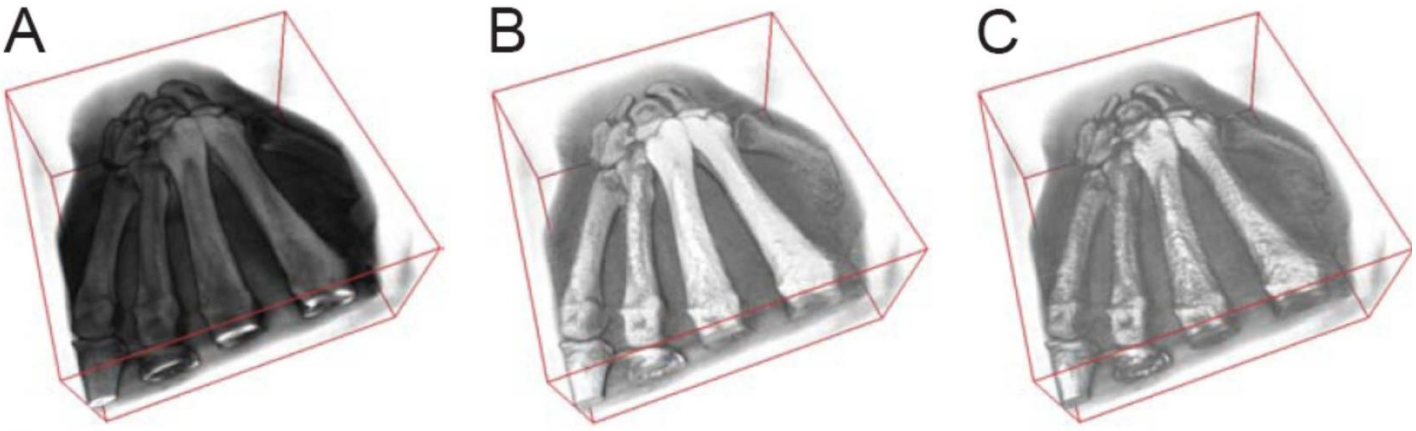
$$\nabla f(x,y,z) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$$

  - Bei Voxeldaten Diskretisierung erforderlich:

$$\frac{\partial f}{\partial x} \approx \frac{f(x+1,y,z) - f(x,y,z)}{1} \qquad \text{(Vorwärtsdifferenz)}$$

$$\frac{\partial f}{\partial x} \approx \frac{f(x+1,y,z) - f(x-1,y,z)}{2} \qquad \text{(Zentrale Differenz)}$$

Ray Casting: Beleuchtung



(A) Ohne Beleuchtung (B) Diffuses Licht (C) Spekulares Licht

R. Erfrurt: „Vorintegriertes Volume Rendering: Slicing vs. Raycasting“, Seminar Medizinische Datenverarbeitung SS 2007- Lehr- und Forschungsgebiet Informatik 8 ”Computergrafik und Multimedia”, RWTH Aachen

Ray Casting

Pro

- Einfach zu implementieren
- Liefert sehr gute Ergebnisse

Contra

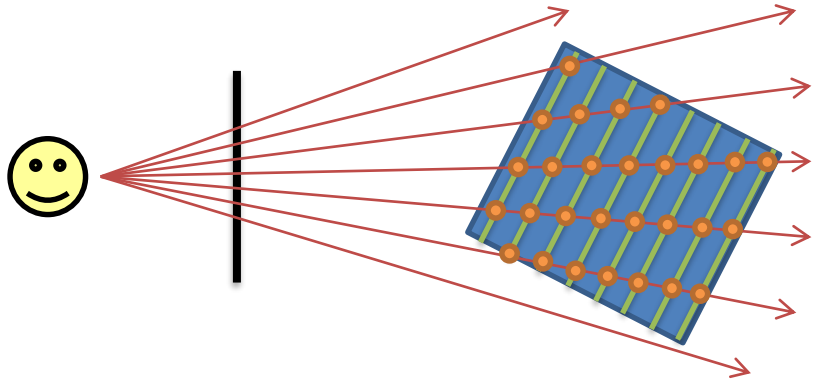
- Sehr rechenintensiv

https://de.wikipedia.org/wiki/Raycasting

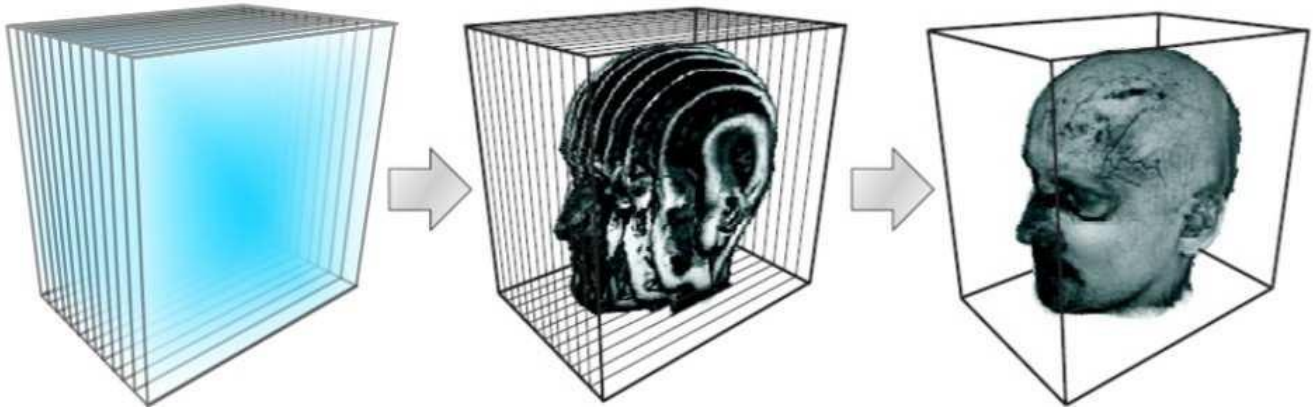
11.3 TEXTURE SLICING

2D Texture Slicing

- Ziel: Optimale Nutzung der GPU (Textur-Speicher) zur Beschleunigung der Berechnung
- Vorgehensweise
  - Zerteilung des Volumens entlang der Hauptachsen
  - Stapel von 2D Schichten → Speichern als 2D Texturen in der GPU
  - Automatische bilineare Interpolation



2D Texture Slicing

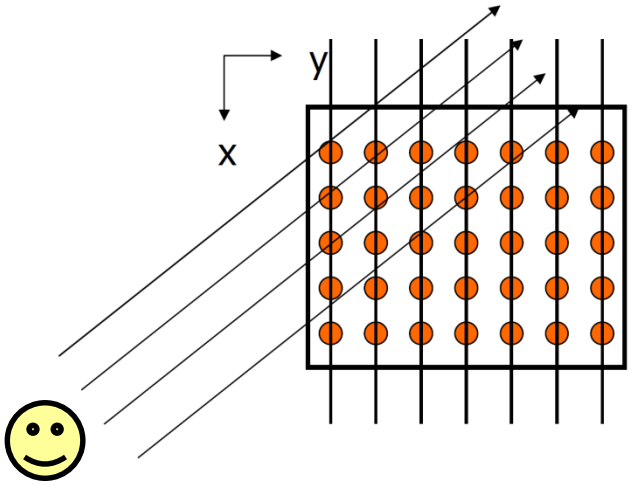


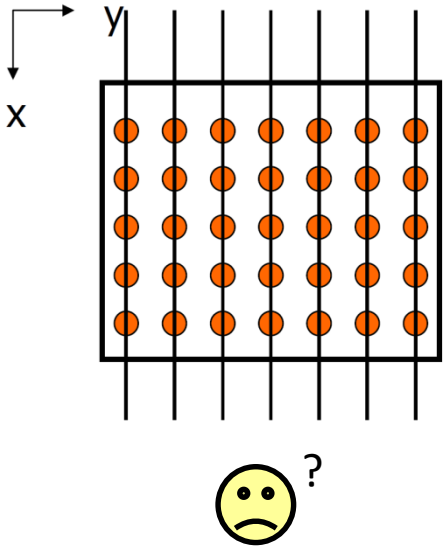
1. Unterteilung der Volumendaten in Schichten
2. Augpunkt und Blickrichtung festlegen
3. Definition von texturierten Polygonen parallel zu den Schichten
4. Rendern jeder Schicht als texturiertes Polygon
5. Blending-Funktion bestimmt Aggregation der Schichten

R. Erfrurt: „Vorintegriertes Volume Rendering: Slicing vs. Raycasting“, Seminar Medizinische Datenverarbeitung SS 2007- Lehr- und Forschungsgebiet Informatik 8 ”Computergrafik und Multimedia”, RWTH Aachen

2D Texture Slicing

- Was wenn sich der Blickwinkel ändert?
  - Ändern der Kameraposition oder Drehen der Polygone





➔ Ändern der Orientierung der Schichten!

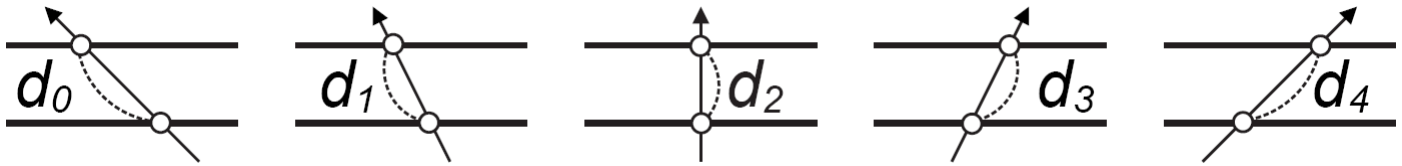
anhand maximaler Komponente des Sichtvektors, z.B. x maximal → yz-Ebene

- Oft werden für alle drei Schichtungsrichtungen texturierte Polygone im Speicher gehalten.

R. Gravis, Ohio State University, 2003

2D Texture Slicing

- Fester Abstand der 2D Schichten: Abtastrate ist bei perspektivischer Projektion nicht konstant über das Volumen
- Abtastrate verringert sich mit Abstand von der Hauptsichtachse



➔ Volumen wird in diesen Bereichen zu hell dargestellt

ACM Siggraph. Real-Time Volume Graphics, Los Angeles, California USA, 2004.

2D Texture Slicing

- Im Gegensatz zu Raycasting: Object-order-Verfahren
  - Ausgehend vom Objekt wird errechnet was dargestellt wird

Pro:

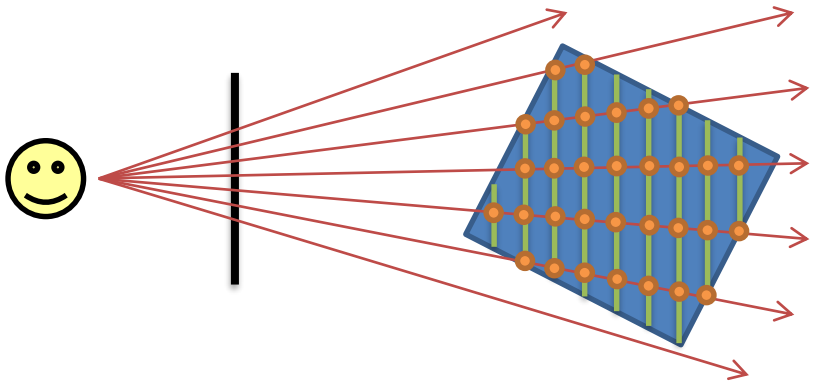
- Schnell
- Auf allen GPU hardware-beschleunigt

Contra:

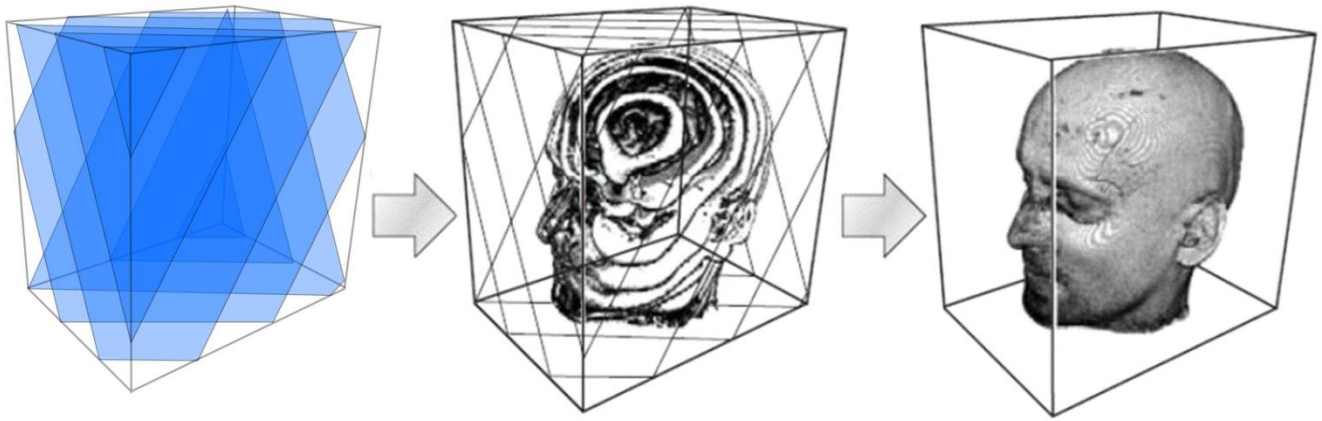
- Mäßige Ausgabequalität
- Artefaktbildung bei hohen Blickwinkeln
- Dreifache Datenmenge für alle Schichtungsrichtungen im Speicher

3D Texture Slicing

- Vorgehensweise:
  - Speichern einer dreidimensionalen Textur in der GPU
  - Berechnung von Texture Slices parallel zur Bildebene
  - Automatische **trilineare** Interpolation



3D Texture Slicing



1. Laden des Volumens in 3D Textur
2. Anzahl der Schichten festlegen, gängigerweise Voxelauflösung
3. Augpunkt und Blickrichtung festlegen
4. Polygone aus dem Volumen „schneiden“, Textur generieren, Orientierung festlegen
5. Rendern jeder Schicht als Polygon von der letzten zur ersten, Blending-Funktion bestimmt Aggregation der Schichten

R. Erfrurt: „Vorintegriertes Volume Rendering: Slicing vs. Raycasting“, Seminar Medizinische Datenverarbeitung SS 2007- Lehr- und Forschungsgebiet Informatik 8 ”Computergrafik und Multimedia”, RWTH Aachen

3D Texture Slicing

Pro:

- Gute Ergebnisse
- Keine Artefakte, unabhängig vom Blickwinkel

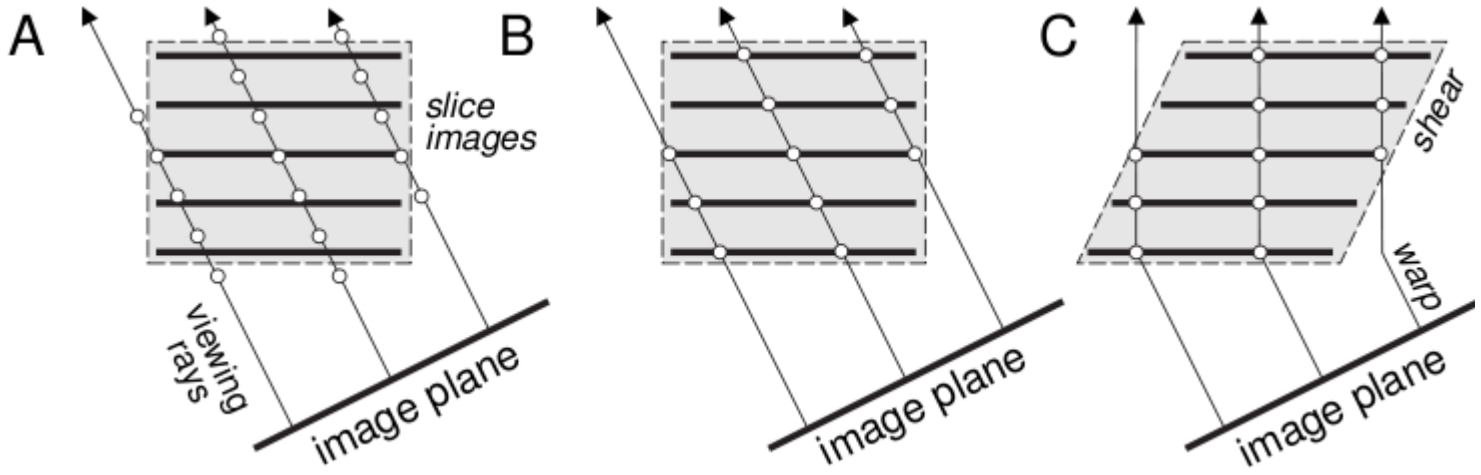
Contra:

- Langsamer, aufwändiger als 2D Texture Mapping
- Bei Blickwinkeländerung Neuberechnung der Schichten erforderlich
- Nicht auf allen GPU hardware-beschleunigt

11.4 SHEAR-WARP RENDERING

Shear-Warp-Rendering

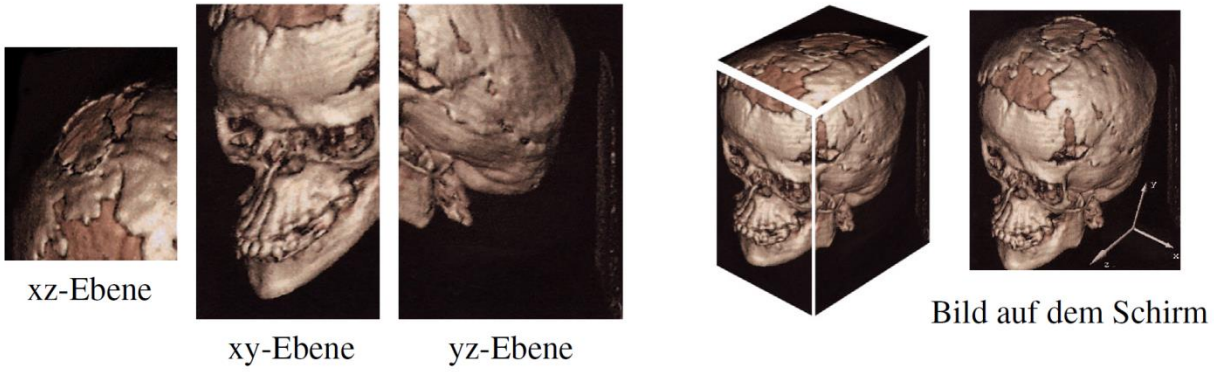
- Schneller Ansatz zum Berechnen des Volume Rendering Integrals, Alternative zu Ray Casting
- Projektion des gesamten Volumens auf die Bildebene
  - bilineare Interpolation ausreichend



H. Jänicke: Visualisierung II, 2. Volumenvisualisierung, Universität Heidelberg

Shear-Warp-Rendering (orthografisch)

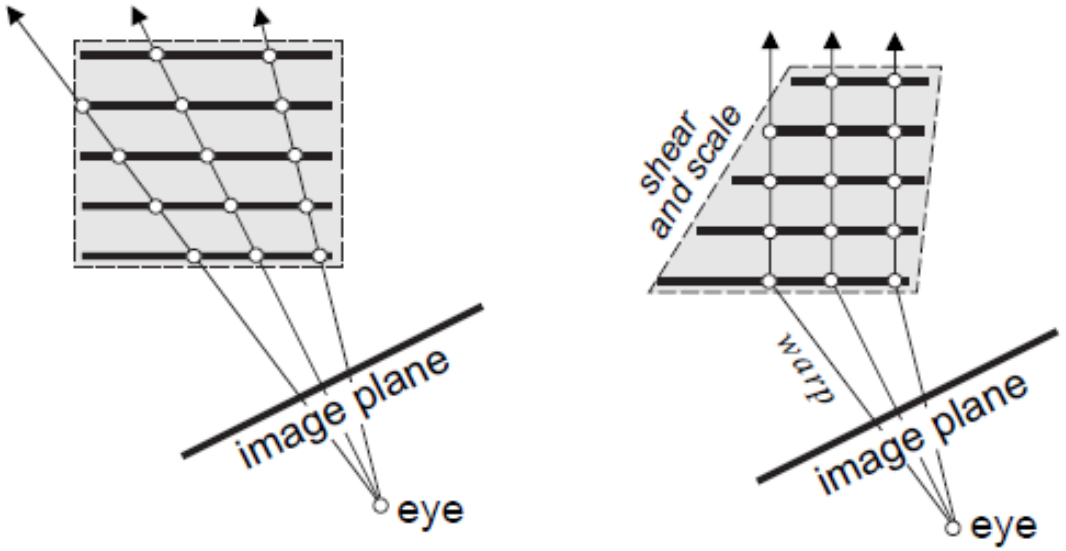
1. Scherung anhand des Winkels der Strahlen zum Volumen (shear)
   - 2D Resampling der Schichten
2. Projektion auf Basisebene (Parallele zum Volumen)
   - Durch das Scheren sind die Projektionsstrahlen orthogonal zur Basisebene!
   - Projiziertes Bild auf die Basisebene ist verzerrt
3. Verzerrung der Basisebenen auf die Bildebene (warp)
   - Orthographische Projektion: Zwischenbilder auf die drei dem Betrachter zugewandten Seiten eines zum Volumen achsenparallelen Quaders in den Proportionen der Bilddaten



S. Krömker: Computergrafik I, Universität Heidelberg

Shear-Warp-Rendering (perspektivisch)

- Perspektivische Projektion: Verkürzung weiter entfernter Schichtbilder muss beim Scheren berücksichtigt werden
- Bei Projektion der Basisebenen auf die Bildebene: Verwendung eines perspektivisch erscheinenden Quaders

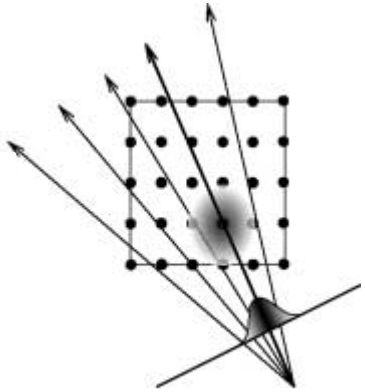


H. Jänicke: Visualisierung II, 2. Volumenvisualisierung, Universität Heidelberg

11.5. SPLATTING

Splatting

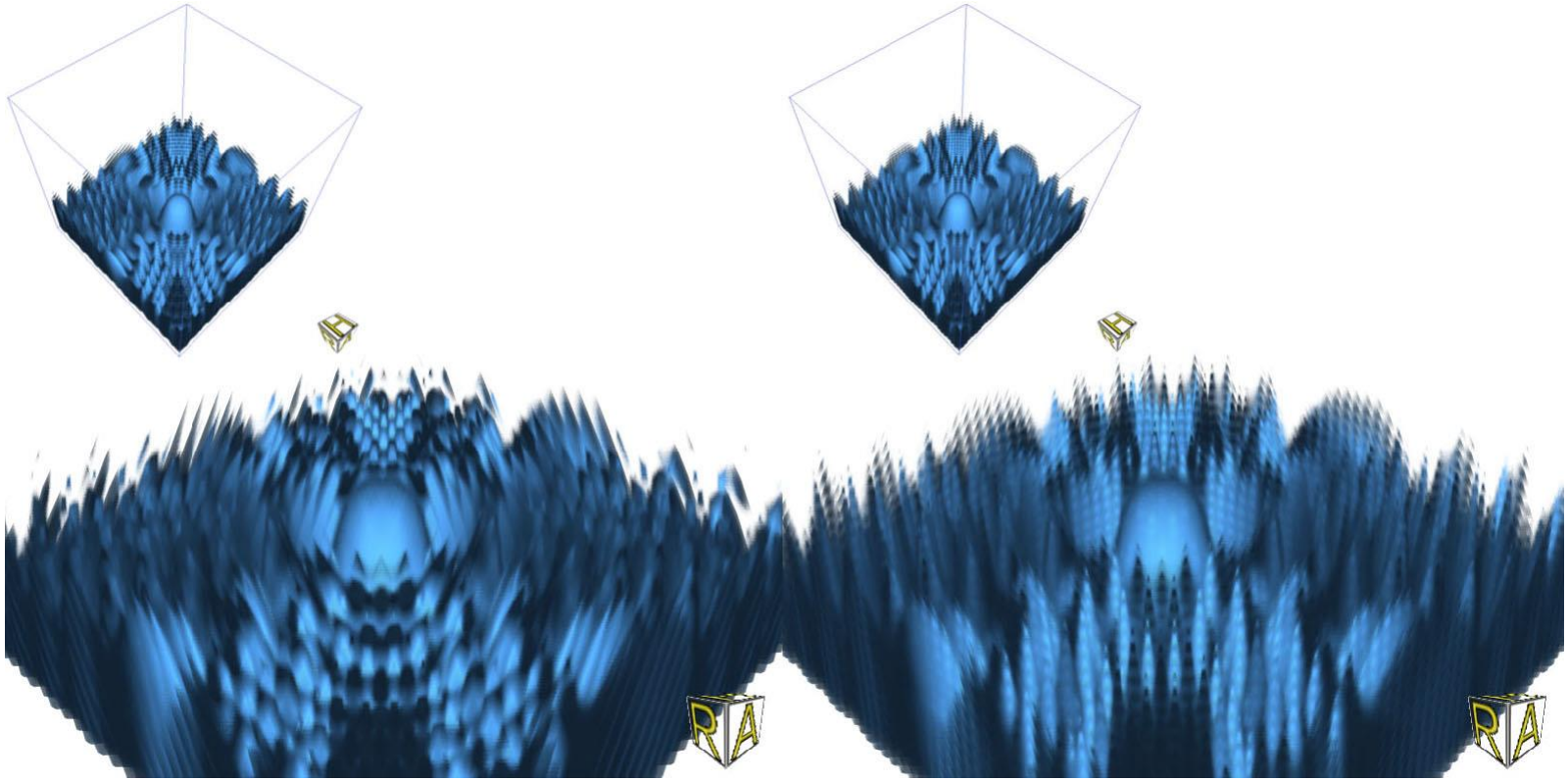
- Object-order-Methode: Jedes Voxel des Volumens wird auf Bildebene projiziert
  - Voxel = 3D Rekonstruktionsfilter
  - Integration der Rekonstruktionsfilter entlang eines Sichtstrahls



- Praktische Umsetzung:
  - Vor-Integration des 3D Rekonstruktionsfilters in Bildebene
    → 2D Rekonstruktionsfilter
  - Speichern des 2D Rekonstruktionsfilters im Textur-Speicher
  - Gewichtung der Textur mit Farbe und Opazität (Transferfunktion)
  - Aufbringen der Textur auf ein Proxypolygon (2D) an der Position des Voxels, parallel zur Bildebene
  - Aufsummieren der 2D Rekonstruktionsfilter durch Alpha-Blending

H. Pfister: „Moderne Volumenvisualisierung“, it – Information Technology 46 (2004) 3

Splatting



3D Texture Slicing

Volume Splatting

F. Vega-Higuera, P. Hastreiter, R. Naraghi, Fahlbusch R., and G. Greiner. High Performance Volume Splatting for Visualization of Neurovascular Data. 2006.

Splatting

Pro

- Geringer Speicheraufwand
- Verschiedene Rekonstruktionsfilter möglich
- Gute Wiedergabe hoher Frequenzen

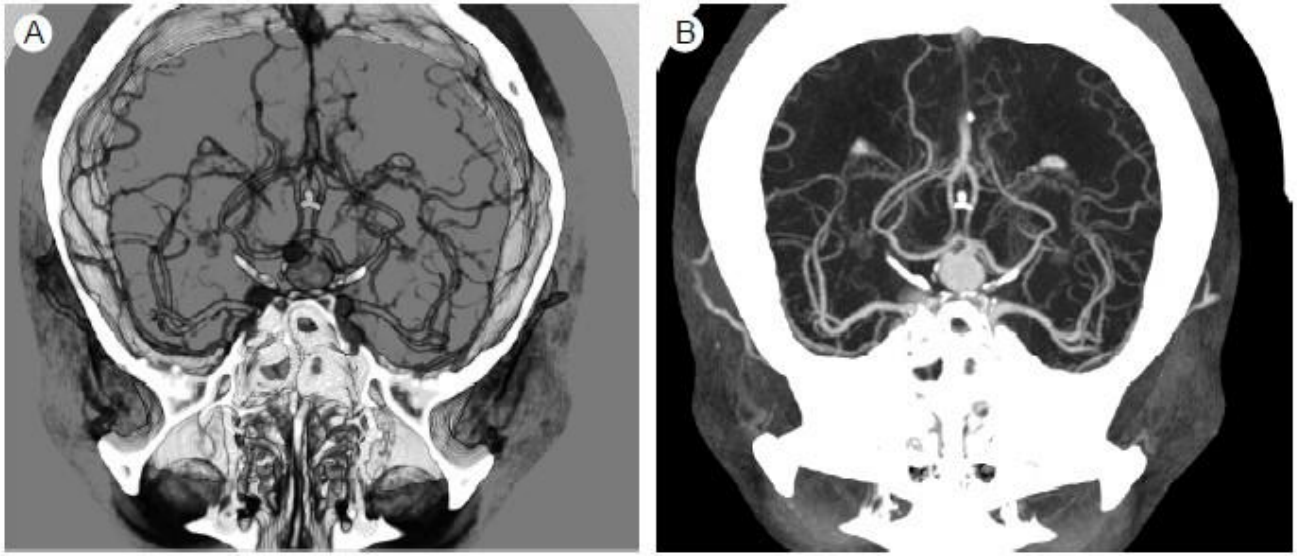
Contra

- Sehr viele Punkte müssen bearbeitet werden da jedes Pixel betrachtet wird → starke Belastung der Geometrieeinheit

11.6 MAXIMUM INTENSITY PROJECTION

Maximum Intensity Projection (MIP)

- Spezielle Art der direkten Volumenvisualiserung
- Statt lösen des Volume Rendering Integrals: höchster Intensitätswert entlang des Strahls



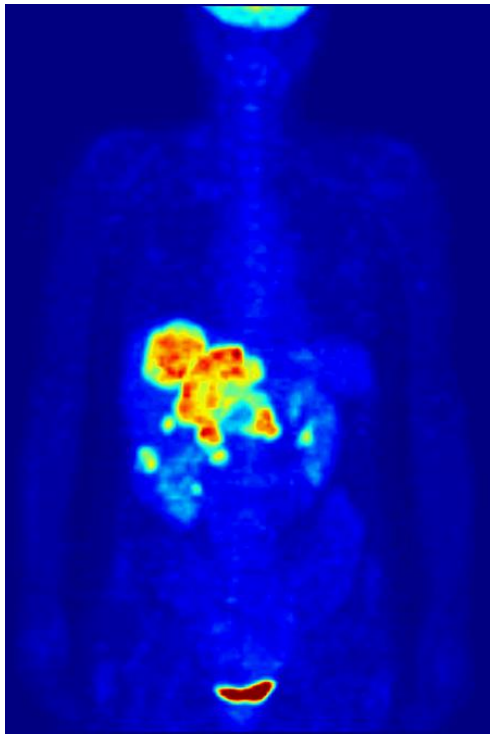
(a) Volume Rendering (b) Maximum Intensity Projection

H. Jänicke: Visualisierung II, 2. Volumenvisualisierung, Universität Heidelberg

Maximum Intensity Projection (MIP)

- Vor allem in der Medizin und Biologie verwendet, da Daten sehr verrauscht sind
- Gute Darstellung z.B. von Gefäßen in der CT-Angiografie, z.B. von Kontrastmittelsubtraktionen, PET-Aufnahmen
- Fehlender Tiefeneindruck
- Drei-dimensionaler Eindruck entsteht oft erst durch Rotation/Änderung der Projektionsrichtung

- Analog gibt es eine Minimum Intensity Projection



https://de.wikipedia.org/wiki/Maximumintensit%C3%A4tsprojektion

ZUSAMMENFASSUNG

Zusammenfassung

- Direkte Volumenvisualisierung:
  - Sichtstrahlen von jedem Pixel der Bildebene durch das Volumen
- Basis: Volume Rendering Integral $C = \int_0^\infty c(t) \cdot e^{-\int_0^d \kappa(\hat{t})\, d\hat{t}}\, dt$
- Ray Casting:
  - Approximation des Volume Rendering Integrals durch Diskretisierung
  - Berechnung durch Alpha Blending (Front-to-Back, Back-to-Front)
  - Transferfunktion: Intensitätswert → Farbwert (Prä- vs. Postklassifikation)
  - Beleuchtung: Normalenberechnung aus Gradient des Volumens
- 2D Texture Slicing / 3D Texture Slicing: Nutzung des Texturspeichers der GPU zur Beschleunigung der Interpolation
- Shear-Warp-Rendering: Scherung der Texturschichten Projektion auf Bildebene
- Splatting: 2D Rekonstruktionsfilter pro Voxel: Aufsummieren durch Alpha-Blending
- Maximum Intensity Projection: höchster Intensitätswert entlang des Strahls

ÜBUNGSAUFGABEN

Übungsfragen Kapitel 11

- Was ist der Unterschied zwischen direkter und indirekter Volumenvisualisierung?
- Was ist Prä- bzw. Post-Klassifizierung bei der Transferfunktion?